Assignment - 1

Analysis on Google-Playstore Dataset

Name- Haridas Bhoite @ Board Infinity

In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
```

In [3]:
```
import matplotlib.pyplot as plt
%matplotlib inline
```

In [4]:
```
import seaborn as sns
```

In [5]:
```
df=pd.read_csv('playstore-analysis (2) (1).csv')
```

In [6]:
```
df.head()
```

Out[6]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19000.0	10,000+	Free	0	Everyone
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14000.0	500,000+	Free	0	Everyone
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8700.0	5,000,000+	Free	0	Everyone
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25000.0	50,000,000+	Free	0	Teen
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2800.0	100,000+	Free	0	Everyone

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10841 entries, 0 to 10840
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   App             10841 non-null  object 
 1   Category        10841 non-null  object 
 2   Rating          9367 non-null   float64
 3   Reviews         10841 non-null  object 
 4   Size            10841 non-null  float64
 5   Installs        10841 non-null  object 
 6   Type            10840 non-null  object 
 7   Price           10841 non-null  object 
 8   Content Rating  10840 non-null  object 
 9   Genres          10841 non-null  object 
 10  Last Updated    10841 non-null  object 
 11  Current Ver     10833 non-null  object 
 12  Android Ver     10838 non-null  object 
dtypes: float64(2), object(11)
memory usage: 1.1+ MB
```

In [8]:
```
df.isnull().sum()
```

Out[8]:
```
App                0
Category           0
Rating          1474
Reviews            0
Size               0
Installs           0
Type               1
Price              0
Content Rating     1
Genres             0
Last Updated       0
Current Ver        8
Android Ver        3
dtype: int64
```

Task:1 Data clean up – Missing value treatment

a. Drop records where rating is missing since rating is our target/study variable

In [9]:
```
df.dropna(how='any', subset=['Rating'], axis=0, inplace = True)
```

In [10]:
```
df.Rating.isnull().sum()
```

Out[10]:
```
0
```

b. Check the null values for the Android Ver column.

i. Are all 3 records having the same problem?

In [11]:
```
df.loc[df['Android Ver'].isnull()]
```

Out[11]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price
4453	[substratum] Vacuum: P	PERSONALIZATION	4.4	230	11000.000000	1,000+	Paid	$1.49
4490	Pi Dark [substratum]	PERSONALIZATION	4.5	189	2100.000000	10,000+	Free	0
10472	Life Made WI-Fi Touchscreen Photo Frame	1.9	19.0	3.0M	21516.529524	Free	0	Everyone

Yes, all 3 records are having same problem ie all are NaN.

ii.Drop the 3rd record i.e. record for "Life Made WIFI …"

In [12]:
```
df.drop([10472], inplace = True)
```

In [13]:
```
df.loc[df['Android Ver'].isnull()]
```

Out[13]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating
4453	[substratum] Vacuum: P	PERSONALIZATION	4.4	230	11000.0	1,000+	Paid	$1.49	Everyone
4490	Pi Dark [substratum]	PERSONALIZATION	4.5	189	2100.0	10,000+	Free	0	Everyone

iii. Replace remaining missing values with the mode

In [14]:
```
df['Android Ver'].fillna(df['Android Ver'].mode()[0], inplace=True)
```

In [15]:
```
df['Android Ver']
```

Out[15]:
```
0              4.0.3 and up
1              4.0.3 and up
2              4.0.3 and up
3                4.2 and up
4                4.4 and up
                  ...
10834            4.1 and up
10836            4.1 and up
10837            4.1 and up
10839    Varies with device
10840    Varies with device
Name: Android Ver, Length: 9366, dtype: object
```

c. Current ver – replace with most common value

In [16]:
```
df['Current Ver'].fillna(df['Current Ver'].mode()[0], inplace=True)
```

Task: 2. Data clean up – correcting the data types

a. Which all variables need to be brought to numeric types?

Reviews and installs need to be brought to numeric types.

b. Price variable -remove $ sign and convert to float

In [17]:
```
df['Price'] = df['Price'].str.replace('$', '').astype(float)
```

In [18]:
```
df.drop(labels=df[df['Price']=='Everyone'].index, inplace = True)
```

In [ ]:
```

```

c. Installs – remove ‘,’ and ‘+’ sign, convert to integer

In [19]:
```
df['Installs'] = df['Installs'].str.replace(',', '')
df['Installs'] = df['Installs'].str.replace('+', '')
df['Installs'] = df['Installs'].astype(int)
```

In [20]:
```
df['Installs'].dtype
```

Out[20]:
```
dtype('int32')
```

d. Convert all other identified columns to numeric

In [21]:
```
df['Reviews']= df['Reviews'].astype('int')
```

In [22]:
```
df['Reviews'].dtype
```

Out[22]:
```
dtype('int32')
```

Task 3. Sanity checks – check for the following and handle accordingly

a. Avg. rating should be between 1 and 5, as only these values are allowed on the play store.

i. Are there any such records? Drop if so.

In [23]:
```
df.loc[df.Rating < 1] & df.loc[df.Rating > 5]
```

Out[23]:

App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Curre V

There are no such records with rating less than 1 or greater than 5.

b. Reviews should not be more than installs as only those who installed can review the app.

i. Are there any such records? Drop if so.

In [24]:
```
df.loc[df['Reviews'] > df['Installs']]
```

Out[24]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Ger
2454	KBA-EZ Health Guide	MEDICAL	5.0	4	25000.000000	1	Free	0.00	Everyone	Mec
4663	Alarmy (Sleep If U Can) - Pro	LIFESTYLE	4.8	10249	21516.529524	10000	Paid	2.49	Everyone	Lifes
5917	Ra Ga Ba	GAME	5.0	2	20000.000000	1	Paid	1.49	Everyone	Arc
6700	Brick Breaker BR	GAME	5.0	7	19000.000000	5	Free	0.00	Everyone	Arc
7402	Trovami se ci riesci	GAME	5.0	11	6100.000000	10	Free	0.00	Everyone	Arc
8591	DN Blog	SOCIAL	5.0	20	4200.000000	10	Free	0.00	Teen	Sc
10697	Mu.F.O.	GAME	5.0	2	16000.000000	1	Paid	0.99	Everyone	Arc

Ans- Yes, there are 7 records where Review is greater than Installs.

In [25]:
```
temp= df[df['Reviews']>df['Installs']].index
df.drop(labels=temp, inplace=True)
```

In [26]:
```
df.loc[df['Reviews']>df['Installs']]
```

Out[26]:

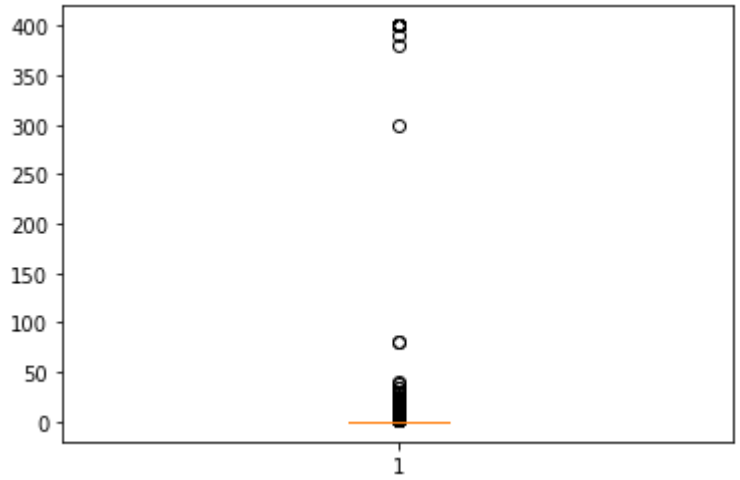
App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Curre \

Task 4. Identify and handle outliers –

a. Price column

i. Make suitable plot to identify outliers in price

In [27]:
```
plt.boxplot(df['Price'])
plt.show()
```



ii.Do you expect apps on the play store to cost $200? Check out these cases

In [28]:
```
print("Yes we expect the apps on the playstore to a cost $200")
```

```
Yes we expect the apps on the playstore to a cost $200
```

In [29]:
```
Price_200=df.loc[df['Price']>200]
Price_200
```

Out[29]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	
4197	most expensive app (H)	FAMILY	4.3	6	1500.0	100	Paid	399.99	Everyone	Ente
4362	💎 I'm rich	LIFESTYLE	3.8	718	26000.0	10000	Paid	399.99	Everyone	
4367	I'm Rich - Trump Edition	LIFESTYLE	3.6	275	7300.0	10000	Paid	400.00	Everyone	
5351	I am rich	LIFESTYLE	3.8	3547	1800.0	100000	Paid	399.99	Everyone	
5354	I am Rich Plus	FAMILY	4.0	856	8700.0	10000	Paid	399.99	Everyone	Ente
5355	I am rich VIP	LIFESTYLE	3.8	411	2600.0	10000	Paid	299.99	Everyone	
5356	I Am Rich Premium	FINANCE	4.1	1867	4700.0	50000	Paid	399.99	Everyone	
5357	I am extremely Rich	LIFESTYLE	2.9	41	2900.0	1000	Paid	379.99	Everyone	
5358	I am Rich!	FINANCE	3.8	93	22000.0	1000	Paid	399.99	Everyone	
5359	I am rich(premium)	FINANCE	3.5	472	965.0	5000	Paid	399.99	Everyone	
5362	I Am Rich Pro	FAMILY	4.4	201	2700.0	5000	Paid	399.99	Everyone	Ente
5364	I am rich (Most expensive app)	FINANCE	4.1	129	2700.0	1000	Paid	399.99	Teen	
5366	I Am Rich	FAMILY	3.6	217	4900.0	10000	Paid	389.99	Everyone	Ente
5369	I am Rich	FINANCE	4.3	180	3800.0	5000	Paid	399.99	Everyone	
5373	I AM RICH PRO PLUS	FINANCE	4.0	36	41000.0	1000	Paid	399.99	Everyone	

iv. Limit data to records with price < $30

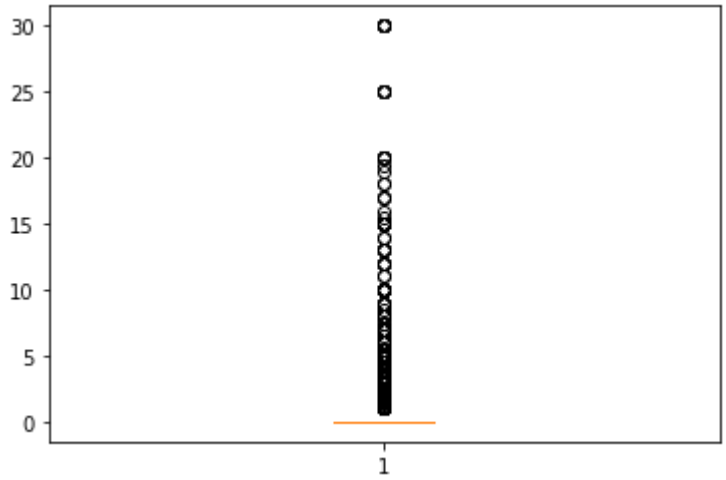
In [30]:
```
less_30 = df[df['Price'] > 30].index
df.drop(labels=less_30, inplace=True)
```

In [31]:
```
count = df.loc[df['Price'] > 30].index
count.value_counts().sum()
```

Out[31]: 0

iii. After dropping the useless records, make the suitable plot again to identify outliers

In [32]:
```
plt.boxplot(df['Price'])
plt.show()
```

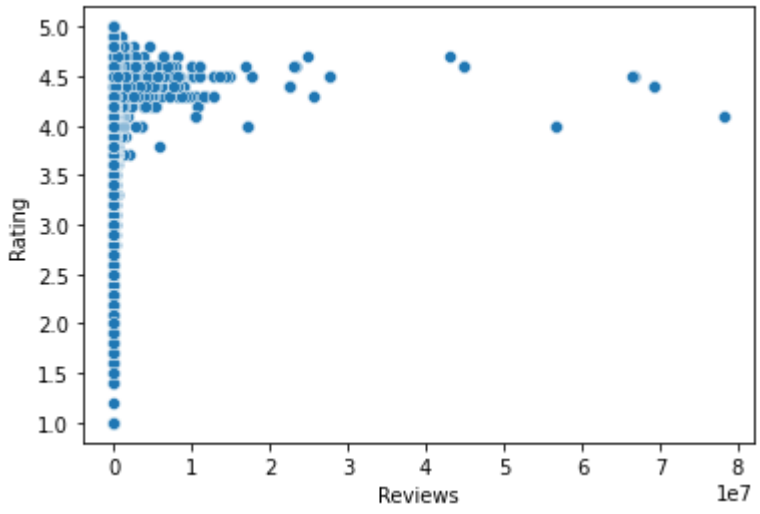


b. Reviews column

i. Make suitable plot

In [33]:
```
sns.scatterplot(df["Reviews"],df["Rating"])
```

Out[33]: <AxesSubplot:xlabel='Reviews', ylabel='Rating'>



ii. Limit data to apps with < 1 Million reviews

In [34]:
```
Rev_grt1m = df[df['Reviews'] > 1000000 ].index
df.drop(labels = Rev_grt1m, inplace=True)
print(Rev_grt1m.value_counts().sum(),'cols dropped')
```

```
704 cols dropped
```

c. Installs

i. What is the 95th percentile of the installs?

In [35]:
```
percentile = df.Installs.quantile(0.95)                    #95th Percentile
 of Installs
print(percentile,"is 95th percentile of Installs")
```

```
10000000.0 is 95th percentile of Installs
```

ii.Drop records having a value more than the 95th percentile

In [36]:
```
temp1 = df[df["Installs"] > percentile].index
df.drop(labels = temp1, inplace = True)
print(temp1.value_counts().sum())#,'cols dropped')
```

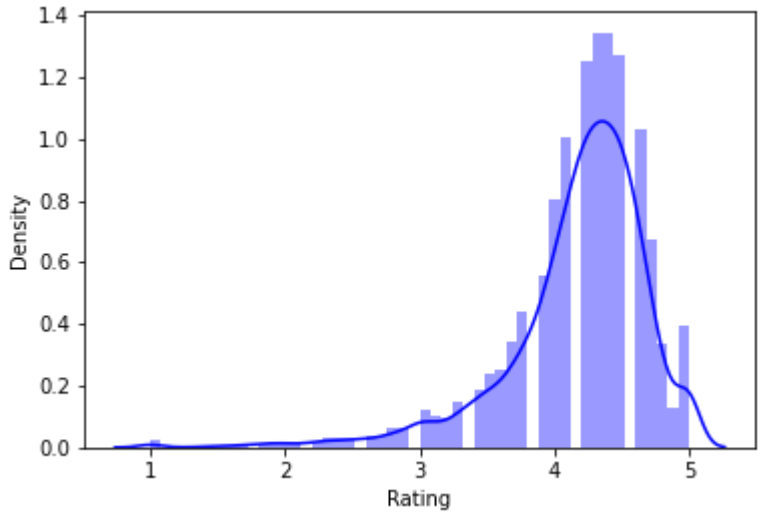
```
199
```

Data analysis to answer business questions

Task 5. What is the distribution of ratings like?(use Seaborn) More skewed towards higher/lower values?

a. How do you explain this?

In [37]:
```
sns.distplot(df['Rating'],color='b')
plt.show()
print('The skewness of this distribution is',df['Rating'].skew())
print('The Median of this distribution {} is greater than mean {} of this dist
ribution'.format(df.Rating.median(),df.Rating.mean()))
```




```
The skewness of this distribution is -1.7434270330647985
The Median of this distribution 4.3 is greater than mean 4.170800237107298 of
this distribution
```

b. What is the implication of this on your analysis?

In [38]:
```
df['Rating'].mode()
```

Out[38]:
```
0    4.3
dtype: float64
```

Since mode>= median > mean, the distribution of Rating is Negatively Skewed.

Therefore distribution of Rating is more Skewed towards lower values.

6. What are the top Content Rating values?

a. Are there any values with very few records?

In [39]:
```
df['Content Rating'].value_counts()
```

Out[39]:
```
Everyone           6782
Teen                900
Mature 17+          417
Everyone 10+        332
Adults only 18+       3
Unrated               1
Name: Content Rating, dtype: int64
```

Adults only 18+ and Unrated are values with very few records so we drop them.

In [40]:
```
#Replacing unwanted values with NaN
cr = []
for k in df['Content Rating']:
    cr.append(k.replace('Adults only 18+','NaN').replace('Unrated','NaN'))

df['Content Rating']=cr
```

In [41]:
```
# Droping the NaN values.
temp2 = df[df["Content Rating"] == 'NaN'].index
df.drop(labels=temp2, inplace=True)
print('droped cols',temp2)
```

```
droped cols Int64Index([298, 3043, 6424, 8266], dtype='int64')
```

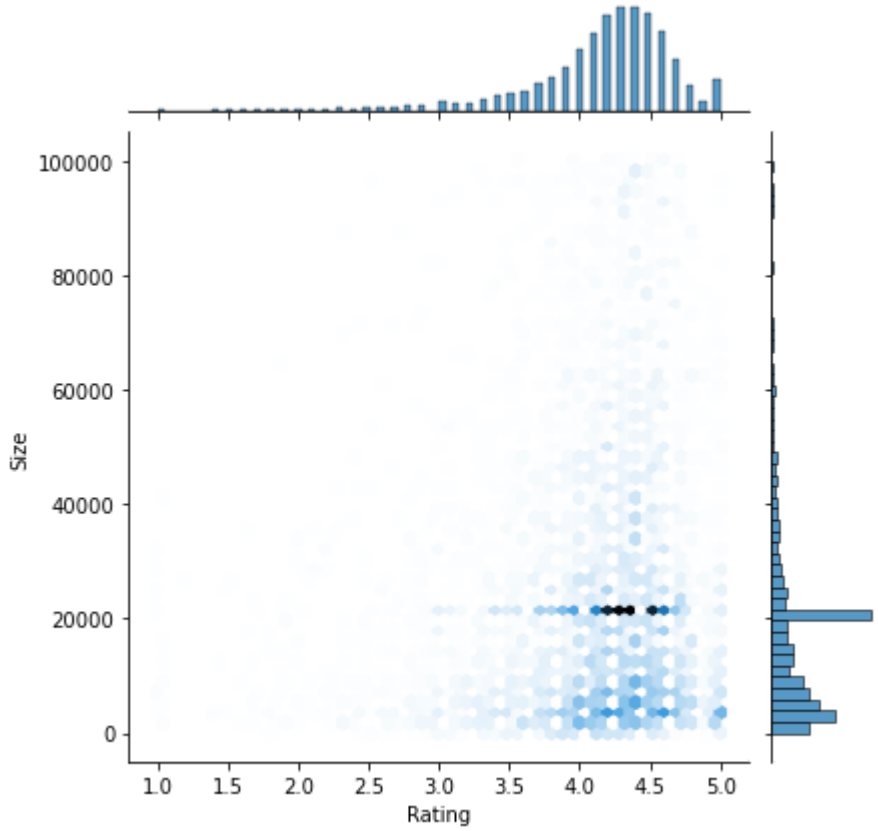
In [42]:
```
df['Content Rating'].value_counts()   # just checking
```

Out[42]:
```
Everyone        6782
Teen             900
Mature 17+       417
Everyone 10+     332
Name: Content Rating, dtype: int64
```

Task 7. Effect of size on rating

a. Make a joinplot to understand the effect of size on rating

In [43]:
```
sns.jointplot(y ='Size', x ='Rating', data = df, kind ='hex')
plt.show()
```



b. Do you see any patterns?

Yes, patterns can be observed between Size and Rating ie. their is correlation between Size and Rating.

c. How do you explain the pattern?

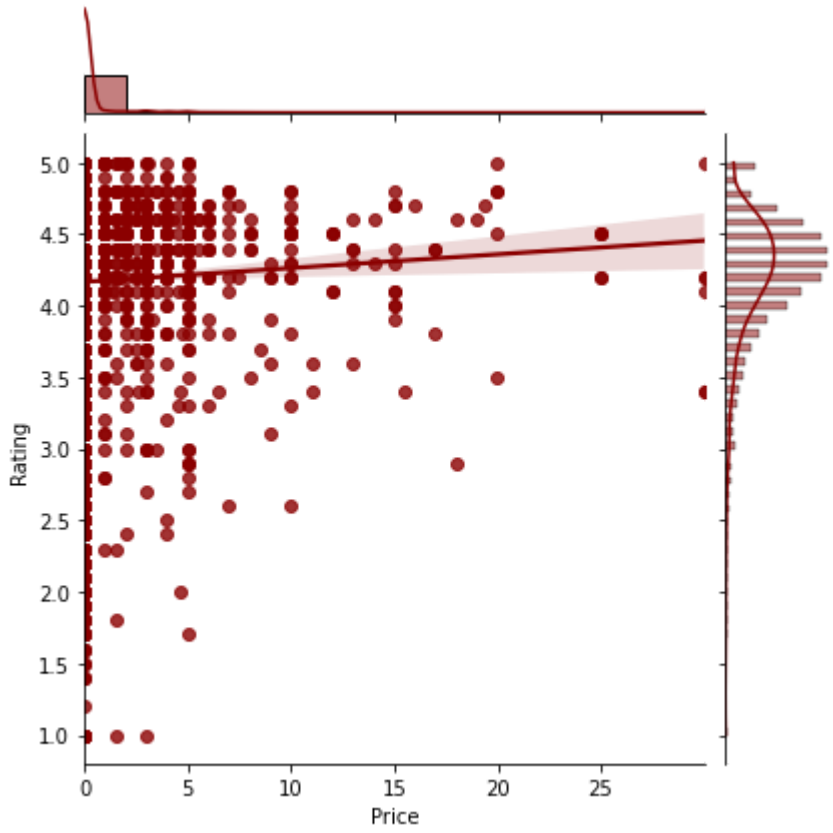
Generally on increasing Rating, Size of App also increases. But this is not always true ie. for higher Rating, their is

constant Size. Thus we can conclude that their is positive correlation between Size and Rating.

Task 8. Effect of price on rating

a. Make a jointplot (with regression line)

In [44]:
```
sns.jointplot(x='Price', y='Rating', data=df, kind='reg',color='darkred')
plt.show()
```



b. What pattern do you see?

Generally on increasing the Price, Rating remains almost constant greater than 4.

c. How do you explain the pattern?

Since on increasing the Price, Rating remains almost constant greater than 4. Thus it can be concluded that their is very weak Positive correlation between Rating and Price.

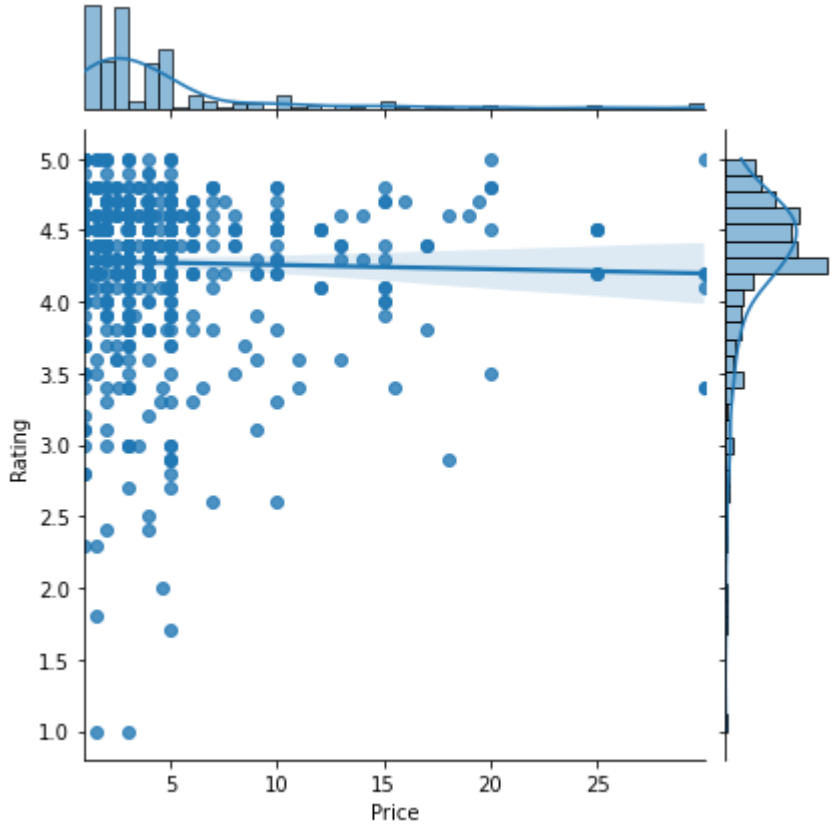
In [45]:
```
df.corr()
```

Out[45]:

	Rating	Reviews	Size	Installs	Price
Rating	1.000000	0.158547	0.058076	0.118414	0.031479
Reviews	0.158547	1.000000	0.204667	0.736038	-0.073446
Size	0.058076	0.204667	1.000000	0.190741	-0.001054
Installs	0.118414	0.736038	0.190741	1.000000	-0.110507
Price	0.031479	-0.073446	-0.001054	-0.110507	1.000000

d. Replot the data, this time with only records with price > 0

In [46]:
```
df1=df.loc[df.Price>0]
sns.jointplot(x='Price', y='Rating', data=df1, kind='reg')
plt.show()
```



e. Does the pattern change?

Yes, On limiting the record with Price > 0, the overall pattern changed a slight

i.e their is very weakly Negative Correlation between Price and Rating.

In [47]:
```
df1.corr()
```

Out[47]:

	Rating	Reviews	Size	Installs	Price
Rating	1.000000	0.095986	0.117943	0.063960	-0.025975
Reviews	0.095986	1.000000	0.163959	0.787628	-0.049764
Size	0.117943	0.163959	1.000000	0.119255	0.024912
Installs	0.063960	0.787628	0.119255	1.000000	-0.057710
Price	-0.025975	-0.049764	0.024912	-0.057710	1.000000

f. What is your overall inference on the effect of price on the rating

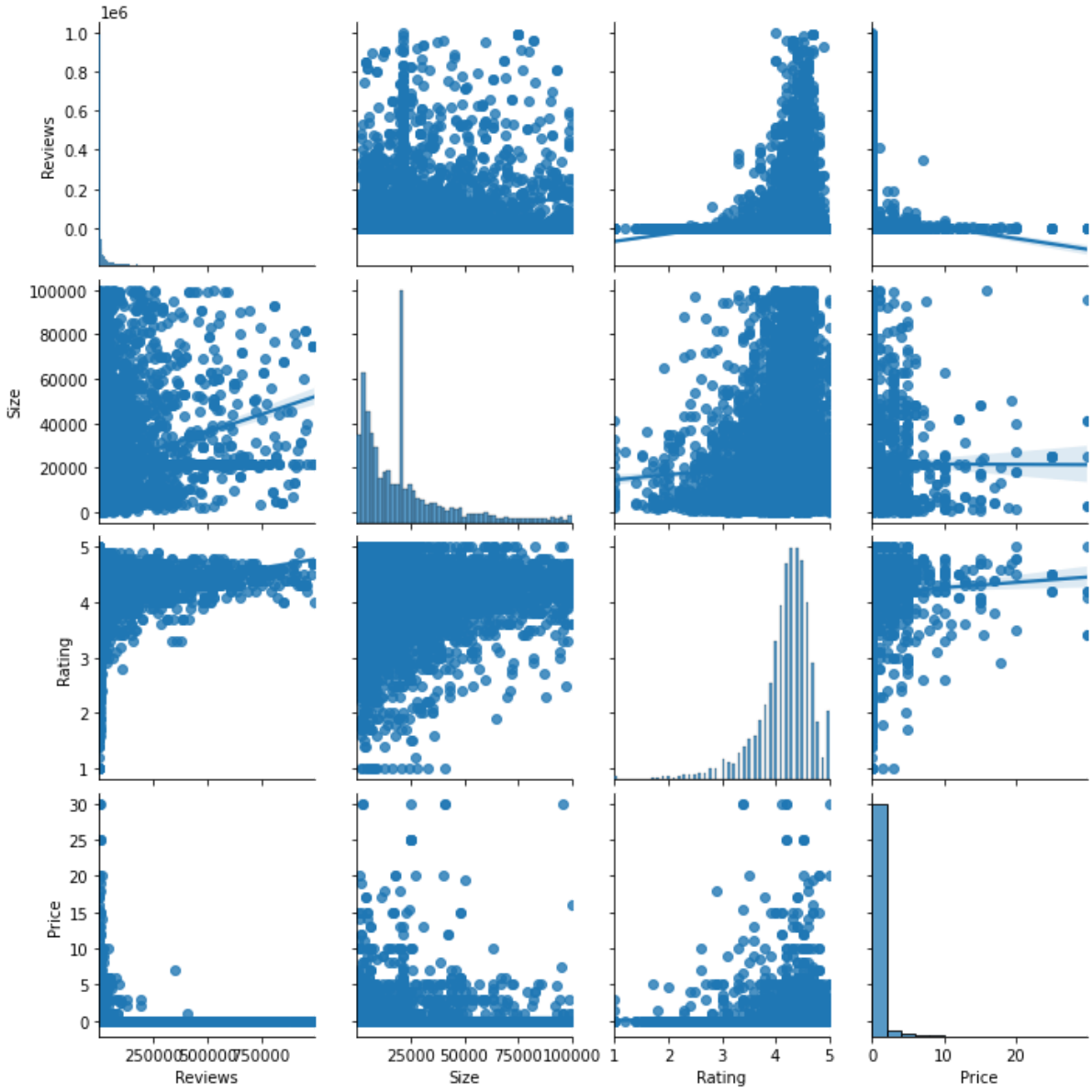
Generally increasing the Prices, doesn't have signifcant effect on Higher Rating.

For Higher Price, Rating is High and almost constant ie greater than 4

9. Look at all the numeric interactions together –

a. Make a pairplort with the colulmns - 'Reviews', 'Size', 'Rating', 'Price'

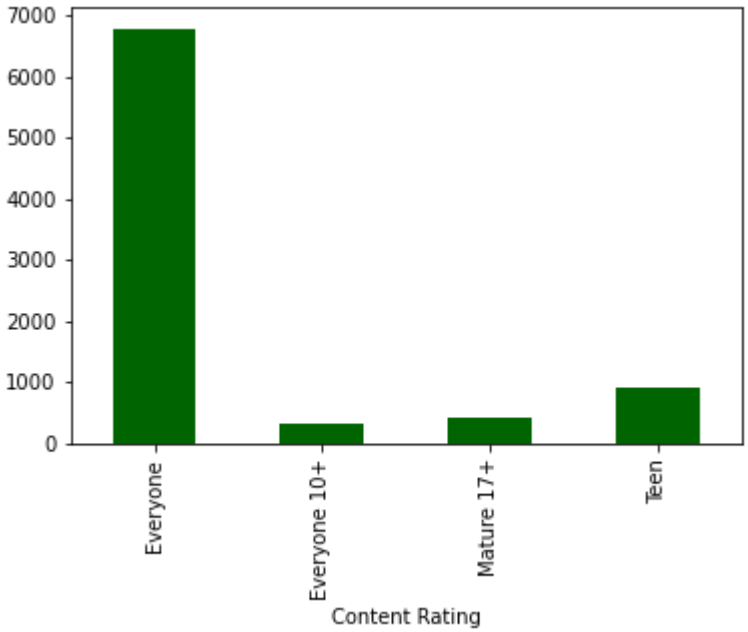
In [48]:
```
sns.pairplot(df, vars=['Reviews', 'Size', 'Rating', 'Price'], kind='reg')
plt.show()
```



Task 10. Rating vs. content rating

a. Make a bar plot displaying the rating for each content rating

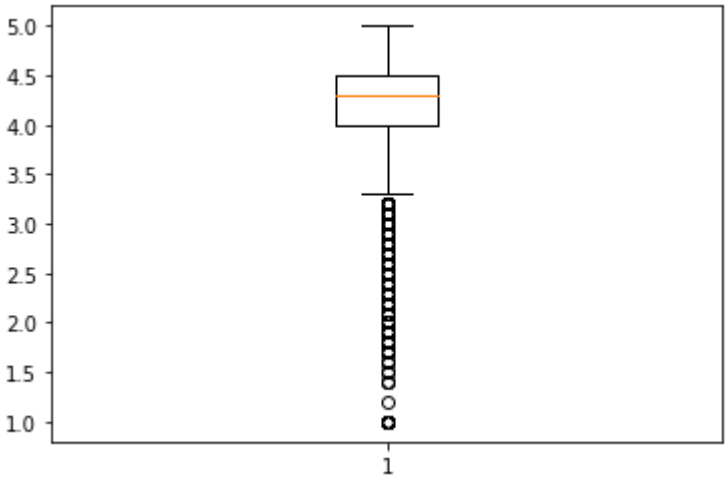
In [49]:
```
df.groupby(['Content Rating'])['Rating'].count().plot.bar(color="darkgreen")
plt.show()
```



b. Which metric would you use? Mean? Median? Some other quantile?

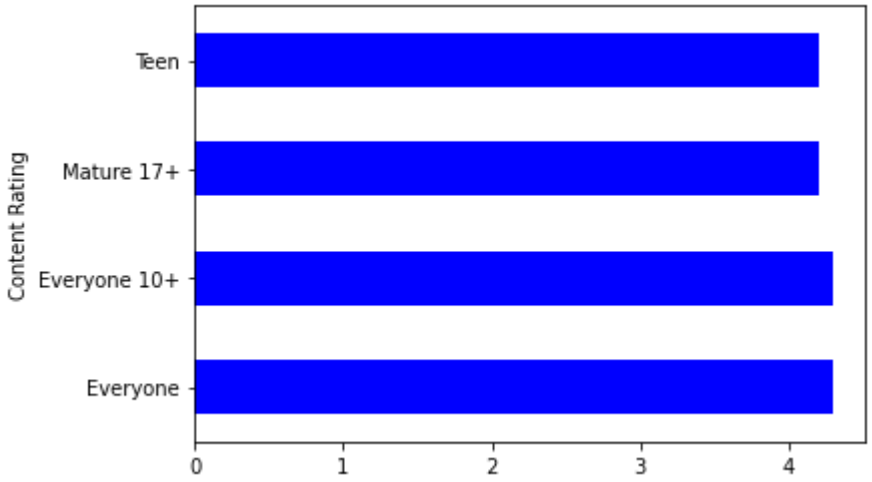
We must use Median in this case as we are having Outliers in Rating. Because in case of Outliers , median is the best measure of central tendency.

In [50]:
```
plt.boxplot(df['Rating'])
plt.show()
```



c. Choose the right metric and plot

In [51]:
```
df.groupby(['Content Rating'])['Rating'].median().plot.barh(color='blue')
plt.show()
```



Task 11. Content rating vs. size vs. rating – 3 variables at a time

a. Create 5 buckets (20% records in each) based on Size

In [52]:
```
bins=[0, 20000, 40000, 60000, 80000, 100000]
df['Bucket Size'] = pd.cut(df['Size'], bins, labels=['0-20k','20k-40k','40k-60k','60k-80k','80k-100k'])
pd.pivot_table(df, values='Rating', index='Bucket Size', columns='Content Rating')
```

Out[52]:

Content Rating	Everyone	Everyone 10+	Mature 17+	Teen
Bucket Size				
0-20k	4.145730	4.247561	4.010582	4.182240
20k-40k	4.200195	4.169811	4.156291	4.170432
40k-60k	4.167083	4.263636	4.190476	4.237383
60k-80k	4.245408	4.280769	4.200000	4.274194
80k-100k	4.260127	4.304762	4.252632	4.270313

b. By Content Rating vs. Size buckets, get the rating (20th percentile) for each combination

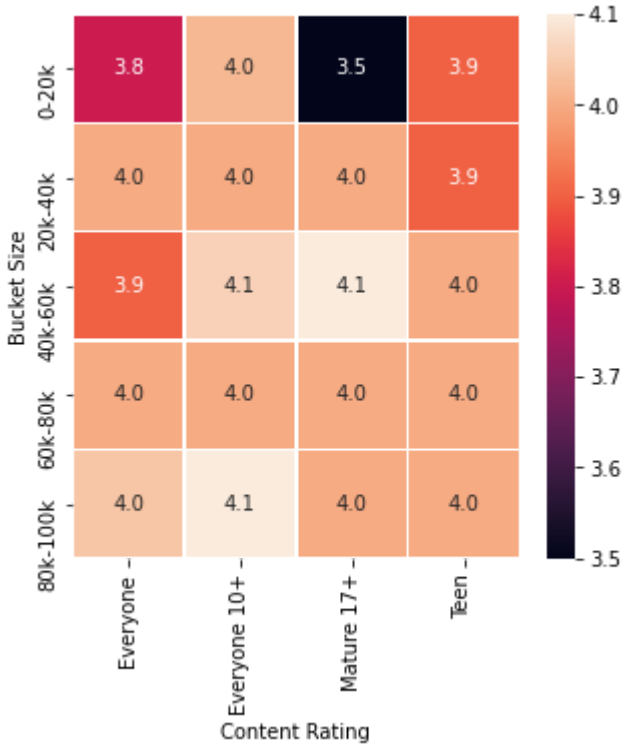
In [53]:
```
temp3=pd.pivot_table(df, values='Rating', index='Bucket Size', columns='Content Rating', aggfunc=lambda x:np.quantile(x,0.2))
temp3
```

Out[53]:

Content Rating	Everyone	Everyone 10+	Mature 17+	Teen
Bucket Size				
0-20k	3.80	4.02	3.5	3.9
20k-40k	4.00	4.00	4.0	3.9
40k-60k	3.90	4.06	4.1	4.0
60k-80k	4.00	4.00	4.0	4.0
80k-100k	4.04	4.10	4.0	4.0

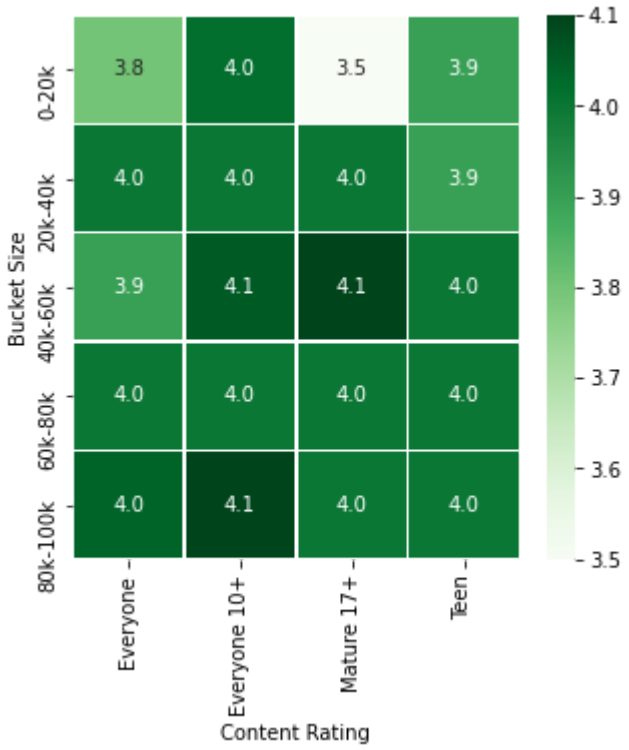
i. Annotated

In [54]:
```
f,ax = plt.subplots(figsize=(5, 5))
sns.heatmap(temp3, annot=True, linewidths=.5, fmt='.1f',ax=ax)
plt.show()
```



ii.Greens color map

In [55]:
```
f,ax = plt.subplots(figsize=(5, 5))
sns.heatmap(temp3, annot=True, linewidths=.5, cmap='Greens',fmt='.1f',ax=ax)
plt.show()
```



d. What's your inference? Are lighter apps preferred in all categories? Heavier? Some?

Based on analysis, its not true that lighter apps are preferred in all categories. Because apps with size 40k-60k and 80k-100k have got the highest rating in all cateegories. So, in general we can conclude that heavier apps are preferred in all categories.

In [56]:
```
# Thank You
```

Submitted by- Haridas Bhoite at Board Infinity

In [ ]:
```

```